Electronic Version
Stylesheet Version v1.1.1

Description

INTRA-FIELD INTERPOLATION METHOD AND APPARATUS

BACKGROUND OF INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to a method and apparatus for processing pixel values, and more specifically, to a method and apparatus of performing intra-field interpolation for generating interpolated pixel values.

[0003] 2. Description of the Prior Art

[0004] Many television and video signals are interlaced, where the set of scan lines (typically 525 for NTSC color television) which make up a single video frame are not scanned or transmitted sequentially. Rather, the video frame is divided into two "fields", each field comprising every other scan line. In television, the scan lines comprising one field are transmitted first, followed by the scan lines of the second field.

[0005] However, a number of display devices, such as computer monitors, are not interlaced. Rather, these devices sequentially scan the entire display area, one scan line after another. To display an interlaced scanned sequence, such as a video signal, on such progressively scanned devices, a deinterlacing process must convert each separate field into a complete display frame that can be sequentially output to the display device. The main task of a deinterlacing process is to reconstruct the missing line between each of the scan lines of an interlaced field.

[0006] There are two primary deinterlacing methods, each with their own strengths and weaknesses. "Inter-field" techniques simply merge the data from the second field with the data from the first field to produce a completed frame. If there is no motion in the video frame, such methods yield an ideal reconstituted picture. Vertical resolution can be as good as an original noninterlaced frame. However, if there is motion within the video signal, motion effects will generally be visible to the human eye. Motion effects arise when an object, which was in one location during the scanning of the first field, has moved when the alternating scan lines of the second field are scanned. Simply combining the interlaced scan lines of the two fields yields an

unacceptable rendition of the object.

[0007] "Intra-field" techniques use data only from a single field to produce a complete frame. Such methods are better suited for video frames having motion. With an intra-field technique, the values for non-existent pixels are interpolated from pixel values in the scan lines above and below the non-existent pixels. While this technique produces no deleterious motion effect, since it does not incorporate motion from one field to the next, it also does not enhance vertical resolution, since it merely interpolates from existing pixel values within a single field and does not use pixel information for missing scan lines from the second field. Also, simple intra-field deinterlacing techniques (such as simple vertical interpolation) tend to generate unacceptable jagged pictures along diagonal edges.

SUMMARY OF INVENTION

[0008] It is therefore one of objectives of the present invention to provide an intra-field interpolation method and apparatus for effectively and accurately generating the interpolated pixel value.

[0009] According to embodiments of the present invention, an intra-field interpolation method for generating a target pixel value is disclosed. The method comprises: receiving

a plurality of pixel values of an image field; generating a first pixel difference set from the received pixel values using a first pixel difference algorithm; generating a second pixel difference set from the received pixel values using a second pixel difference algorithm; and blending the received pixel values according to the first pixel difference set and the second pixel difference set, to generate the target pixel value.

[0010] These and other objectives of the claimed invention will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the embodiments that is illustrated in the various figures and drawings.

Brief Description of Drawings

[0011] Fig.1 is a portion of an image comprising pixels arranged in rows.

[0012] Fig.2 is an intra-field interpolation apparatus according to an embodiment of the present invention.

[0013] Fig.3 is a diagram illustrating a plurality of pixel values in an upper line and a lower line of the pixel to be interpolated.

[0014] Fig.4 is a flowchart of the operation of the angle determining unit 210 according to an embodiment of the

present invention.

[0015] Fig.5 is a flowchart of a weighted blending algorithm adopted by the weighted blending unit 216 according to an embodiment of the present invention.

DETAILED DESCRIPTION

[0016] Please refer to Fig.1. Fig.1 illustrates a portion of an image 100 comprising pixels arranged in rows (or lines.) Pixels 111-119 are arranged in one line 110 of the image 100 and pixels 131-139 are arranged in another line 130 of the image 100. As is well known, the lines 110 and 130 shown in Fig.1 belongs to the same field (e.g., the even field or the odd field) of the image 100. As will be detailed in the following descriptions, by implementing embodiments of the present invention, the resolution of the image is to be enhanced by generating additional pixels between the two lines of pixels on a pixel-by-pixel basis using intra-field interpolation during de-interlacing process. The figure and following discussion describe the generation of additional pixel 125. Please note that, although the intra-field interpolation method and apparatus to be described herein focus on a de-interlacing application, they can also be implemented in other applications wherever suitable.

[0017] Please refer to Fig.2. Fig.2 illustrates an intra-field interpolation apparatus 200, which may be embedded in a de-interlacing engine, according to an embodiment of the present invention. The intra-field interpolation apparatus 200 receives an interlaced input signal, and generates a corresponding de-interlaced output signal. The apparatus 200 comprises a low-pass filtering unit 202, which filters the input signal to reduce noise effect; a first pixel difference unit 206, which adopts a first pixel difference algorithm, and in this embodiment, references a result of a gradient unit 204, to generate a first pixel difference set from the filtered input signal; a second pixel difference unit 208, which adopts a second pixel difference algorithm to generate a second pixel difference set from the filtered input signal; an angle determining unit 210, in this embodiment composed of an angle checking and correcting unit 212 and an angle voting unit 214, which determines an optimal angle according to the results of the first pixel difference unit 206, the second pixel difference unit 208, and/or the angle voting unit 214; and a weighted blending unit 216, which weighted blends, in this embodiment through interpolation, pixel values in the interlaced input signal (e.g., pixel values in the lines 110,

130) to generate de-interlaced output signal (e.g., the additional pixel 125.)

[0018] First of all, the operation of the low-pass filtering unit 202 will be described. The purpose of adopting a low-pass filtering operation before any further processing to the incoming data usually includes noise suppresion or image smoothening. A typical low-pass flitering operation may include, for each incoming pixel, incorporating influence of adjacent pixels thereto, and an example may read as follows:

[0019] low_pass_value = [previous_value + 2 × current_value + next_value]

[0020] wherein low_pass_value is a low-pass filtered version of a currently incoming pixel value (i.e., current_value,) while previous_value, current_value, and next value are three consecutively incoming pixel values (e.g., pixels 111, 112, 113 in Fig.1.) Please note that, as is well known to those of ordinary skill in the art, the above-mentioned low-pass filtering operation serves only as one example, and other low-pass filtering algorithm may be substituted in where the application sees fit.

[0021] For the following descriptions, the convention shown in Fig.3 is to be adopted. The pixel values that is to be input

to the gradient unit 204, the second pixel difference unit 208, and the weighted blending unit 216, are respectively labeled Up[x] and Down[x], wherein x = -4, ..., 0, ..., 4, as shown in Fig.3. In this embodiment, these pixel values may be the low-pass filtered version of the corresponding pixels 111-119 in the line 110 and pixels 131-139 in the line 130.

[0022] In this embodiment, before the operation of the first pixel difference unit 206, the gradient unit 204 first operates to determine if the gradient of pixel values in the upper line (e.g., the line 110) and the gradient of pixel values in the lower line (e.g., the line 130) of a current pixel to be interpolated (e.g., the additional pixel 125) possess certain characteristics. Here such an operation of the gradient unit 204 is implemented by excluding certain situations. The gradient unit 204 first checks to determine the values of a set of parameters GradUp[x], x = -3, ..., 0, ..., 4, each of which is representative of a gradient status between two corresponding pixels in the upper line (e.g., GradUp[-3] corresponds to pixel values Up[-4] and Up[-3], and so on.) In this embodiment, the following operations are performed to determine GradUp[x]:

[0023] If (Up[x] - Up[x-1] > Thl_P)

[0024] GradUp[x] = 1

[0025] Else if (Up[x] - Up[x-1] < Thl_N)

[0026] GradUp[x] = -1

[0027] Else

[0028] GradUp[x] = GradUp[x-1]

[0029] wherein Thl_P and Thl_N are threshold values, with nominal values of 15 and 15. Similar operations can be performed on the pixel values Down[x] to render another set of parameters GradDown[x]. Then, the gradient unit 204 checks the following two situations: either when GradUp[x] or GradDown[x] contains sign toggling among element parameters, or when GradUp[x] and GradDown[x] does not contain sign togggling, but are of the same tendency (i.e., both of all 1s or all 1s,) is a parameter Gradient set to have a value of 0. Otherwise the parameter Gradient is set to have a value of 1.

[0030] Please note that the above-mentioned embodiment gradient unit 204 only serves as an example. It should be appreciated by those of ordinary skill in the art that other algorithms, such as the well-known Sobel algorithm, may be adopted to serve the purpose of the gradient unit 204

in this invention.

[0031] After the value of the parameter Gradient has been determined, the first pixel difference unit 206 will operate to calculate a first pixel difference set, and to determine two angles, one on either side of a normal axis 150, according to the parameter Gradient as well as the first pixel difference set. According to this embodiment, the following pixel difference calculation is performed for each of a plurality of pairs of pixel sets in the upper line 110 and the lower line 130, to generate an element of the first pixel difference set first_diff[x]:

[0032] $$first\_diff[x] = |Up[x+1] - Down[-x+1]| + |Up[x] - Down[-x]| + |Up[x-1] - Down[-x-1]|$$

[0033] wherein x = −3, ..., 0, ..., 3, as an example. The above function of the first pixel difference set reads the sum of absolute differences (SAD) between a pair of pixel sets each in the upper line 110 and in the lower line 130 (e.g., the pixel set 162 with pixels 117, 118, 119, and the pixel set 164 with pixels 131, 132, 133, and so on.) Here the pair of pixel sets resides on a vertual axis 160 through the pixel to be interpolated 125, forming an angle θ with the normal axis 150.

[0034] After all the elements in first_diff[x] has been found, the following operation is performed to determine, on either

side of the normal axis 150, a pair of pixel sets (or a corresponding angle), which results in two angle values coming out of the first pixel difference unit 206. If Gradient = 0, find a pair of pixel sets (i.e., an angle) with the smallest first_diff[x] value, which satisfies the following condition: Min(Up[0], Down[0])

$$\leqq$$

(Up[x] + Down[x])/2

$$\leqq$$

Max(Up[0], Down[0]), wherein Min is the minimum function, and Max is the maximum function; while if Gradient = 1, find a pair of pixel sets (i.e., an angle) with the smallest first_diff[x] value. That is, when the gradient situation of the upper line 110 and the lower line 130 of the pixel to be interpolated 125 is deemed to have certain characteristics (i.e., Gradient = 1), an angle with the smallest first_diff[x] value is selected without any further limitations. Otherwise, an angle with the smallest first_diff[x] is selected with a further boundary limitation, which requires the average of two pixel values on the selected axis (i.e., Up[x] and Down[x]) falls between the pixel values of the

two pixels on the normal axis 150 (i.e., Up[0] and Down[0].) As such, two angle values, θ_1 on the right side (which corresponds to a right pixel difference Right_Pixel_Diff), and θ_2 on the left side (which corresponds to a left pixel difference Left_Pixel_Diff) of the normal axis 150, are determined.

[0035] It is to be noted that although in this embodiment each pixel set in the pair is organized to contain 3 pixels, such number of pixels in the pixel set does not serve as limitation of the invention. Furthermore, the first pixel difference unit 206 adopting a strategy of selecting minimal SAD, partially with a further boundary limitation, also merely serves as an embodiment of the invention, and should not be deemed as limitation.

[0036] Along with the operations of the gradient unit 204 and the first pixel difference unit 206, the second pixel difference unit 208 also operates to generate a second pixel difference set, according to an alternative pixel difference algorithm than the first pixel difference unit 206. In this embodiment, the second pixel difference unit 208 performs the following operation, which may be termed as "reference pixel difference algorithm", to render a second pixel difference set which includes two reference pixel differ-

ences Right_Ref_Diff and Left_Ref_Diff:

[0037] $Ref\_Prev = (Up[-1] + Down[-1])/2$

[0038] $Ref\_Cur = (Up[0] + Down[0])/2$

[0039] $Ref\_Next = (Up[1] + Down[1])/2$

[0040] $Right\_Ref\_Diff = (|Up[0] - Ref\_Prev| + |Down(-2) - Ref\_Prev|) + 2 \times (|Up[1] - Ref\_Cur| + |Down(-1) - Ref\_Cur|) + (|Up[2] - Ref\_Next| + |Down(0) - Ref\_Next|)$

[0041] $Left\_Ref\_Diff = (|Up[-2] - Ref\_Prev| + |Down(0) - Ref\_Prev|) + 2 \times (|Up[-1] - Ref\_Cur| + |Down(1) - Ref\_Cur|) + (|Up[0] - Ref\_Next| + |Down(2) - Ref\_Next|)$

[0042] wherein Right_Ref_Diff indicates an SAD with respect to a plurality of reference pixel values (i.e., Ref_Prev, Ref_Cur, and Ref_Next,) of a pair of pixel sets (here, Up[0] to Up[2] and Down[-2] to Down[0]) along an angle on the right side of the normal axis 150 (here, 45 degrees,) whereas Left_Ref_Diff along an angle on the left side of the normal axis 150 (here, -45 degrees.)

[0043] Also to be noted is that the second pixel difference unit 208 adopting the above-mentioned "reference pixel difference algorithm" serves merely as an embodiment of the invention, and should not be deemed as limitation. Other algorithms, when properly adapted to the purpose of the

second pixel difference unit 208, may also be utilized.

[0044] After the operations of the first pixel difference unit 206 and the second pixel difference unit 208, at least the parameters Right_Pixel_Diff (which corresponds to θ_1,) Left_Pixel_Diff (which corresponds to θ_2,) Right_Ref_Diff, and Left_Ref_Diff are passed to the angle checking and correcting unit 212, and the angle checking and correcting unit 212 will operate, according to these parameters, in conjunction with the operation of the angle voting unit 214, to determine an optimal angle Opt_Angle. The angle voting unit 214 operates to compare an angle at issue together with two stored angles, Pre_Angle and Pre_Pre_Angle, from the checking and correcting operation corresponding to two previous pixels, and it is to be noted that the angles stored in the angle voting unit 214 are those values before the correcting operation by the angle checking and correcting unit 212. Please refer to Fig.4, which illustrates a flowchart of the operation of the angle determining unit 210, and includes the following steps:

[0045] Step 402: Start the flow;

[0046] Step 404: Check to see if the statement |Right_Pixel_Diff Left_Pixel_Diff| < Thl_1 is true, wherein Thl_1 is a thresh-

old value with a nominal value of 80. If it is true, perform Step 406; otherwise perform Step 414. In this embodiment, when the two parameters Right_Pixel_Diff and Left_Pixel_Diff, from the first pixel difference unit 206 are too close to each other, it is considered insufficient to determine the optimal angle Opt_Angle by referencing to the result of the first pixel difference algorithm alone, and results of other additional pixel difference algorithm(s), here, the output of the second pixel difference unit 208, is sought to be resorted to;

[0047] Step 406: Compare the parameters Right_Ref_Diff and Left_Ref_Diff. If Right_Ref_Diff < Left_Ref_Diff, perform Step 408; if Right_Ref_Diff > Left_Ref_Diff, perform Step 410; otherwise perform Step 412;

[0048] Step 408: Perform a first sub-flow, which will be detailed in the following paragraphs;

[0049] Step 410: Perform a second sub-flow, which will be detailed in the following paragraphs;

[0050] Step 412: Perform a third sub-flow, which will be detailed in the following paragraphs;

[0051] Step 414: Compare the parameters Right_Pixel_Diff and Left_Pixel_Diff. If Right_Pixel_Diff < Left_Pixel_Diff, perform Step 416; if Right_Pixel_Diff > Left_Pixel_Diff, per-

form Step 418; otherwise perform Step 420;

[0052] Step 416: Set the optimal angle Opt_Angle = θ_1, which is the one of the two angles on the right side output from the first pixel difference unit 206, and corresponds to the right pixel difference Right_Pixel_Diff;

[0053] Step 418: Set the optimal angle Opt_Angle = θ_2, which is the one of the two angles on the left side output from the first pixel difference unit 206, and corresponds to the left pixel difference Left_Pixel_Diff;

[0054] Step 420: Perform a fourth sub-flow, which will be detailed in the following paragraphs; and

[0055] Step 422: Finish the flow.

[0056] For the first sub-flow mentioned above, the following operations are performed in this embodiment to determine the optimal angle Opt_Angle when Right_Ref_Diff < Left_Ref_Diff in Step 406:

[0057] If (Gradient == 1 && |Right_Ref_Diff - Left_Ref_Diff| < Thl_2 && Pre_Angle == Pre_Pre_Angle == θ_2 [including 0 degree, i.e., the normal direction])

[0058] Opt_Angle = θ_2

[0059] Else

[0060] If (|Right_Ref_Diff - Left_Ref_Diff| < Thl_2 && Pre_Angle

== Pre_Pre_Angle == θ_2 [not including 0 degree, i.e., the normal direction])

[0061] Opt_Angle = θ_2

[0062] Else

[0063] Opt_Angle = θ_1

[0064] wherein Thl_2 is a threshold value with a nominal value of 30. Please note that in this embodiment, the parameter Gradient from the gradient unit 204, the selected angles θ_1, θ_2 by the first pixel difference unit 206, and the parameters Right_Ref_Diff, Left_Ref_Diff are all incorporated into the first sub-flow to determine the optimal angle Opt_Angle. Furthermore, a so-called "voting mechanism" referencing two previous angles together with a current angle (here, θ_2) performed by the angle voting unit 214 is also utilized.

[0065] For the second sub-flow mentioned above, the following operations are performed in this embodiment to determine the optimal angle Opt_Angle when Right_Ref_Diff > Left_Ref_Diff in Step 406:

[0066] If (Gradient == 1 && |Right_Ref_Diff - Left_Ref_Diff| < Thl_2 && Pre_Angle == Pre_Pre_Angle == θ_1 [including 0 degree, i.e., the normal direction])

[0067] Opt_Angle = θ_1

[0068] Else

[0069] If (|Right_Ref_Diff - Left_Ref_Diff| < Thl_2 && Pre_Angle == Pre_Pre_Angle == θ_1 [not including 0 degree, i.e., the normal direction])

[0070] Opt_Angle = θ_1

[0071] Else

[0072] Opt_Angle = θ_2

[0073] Similarly, please note that in this embodiment, the parameter Gradient from the gradient unit 204, the selected angles θ_1, θ_2 by the first pixel difference unit 206, and the parameters Right_Ref_Diff, Left_Ref_Diff are likewise incorporated into the second sub-flow to determine the optimal angle Opt_Angle. Furthermore, a voting mechanism similar to the above-mentioned referencing two previous angles together with a current angle (here, θ_1) performed by the angle voting unit 214 is also utilized.

[0074] For the third sub-flow mentioned above, the following operations are performed in this embodiment to determine the optimal angle Opt_Angle when Right_Ref_Diff = Left_Ref_Diff in Step 406:

[0075] If ($\theta_1 == \theta_2$)

[0076] Opt_Angle = either one of θ_1 and θ_2

[0077] Else

[0078] If (Right_Pixel_Diff < Left_Pixel_Diff)

[0079] Opt_Angle = θ_1

[0080] Else if (Right_Pixel_Diff > Left_Pixel_Diff)

[0081] Opt_Angle = θ_2

[0082] Else

[0083] If (Pre_Angle == θ_1)

[0084] Opt_Angle = θ_1

[0085] If (Pre_Angle == θ_2)

[0086] Opt_Angle = θ_2

[0087] Else

[0088] Opt_Angle = either one of θ_1 and θ_2

[0089] For the fourth sub-flow mentioned above, the following operations are performed in this embodiment to determine the optimal angle Opt_Angle when Right_Pixel_Diff = Left_Pixel_Diff in Step 414:

[0090] If (Pre_Angle == θ_1)

[0091] Opt_Angle = θ_1

[0092] If (Pre_Angle == θ_2)

[0093] Opt_Angle = θ_2

[0094] Else

[0095] Opt_Angle = either one of θ_1 and θ_2

[0096] After the optimal angle Opt_Angle has been determined in the angle determining unit 210, the optimal angle so determined and the pixel difference value (labeled Angle_Pixel_Diff) corresponding to the optimal angle generated by the first pixel difference unit 206 are passed to the weighted blending unit 216 for further operation. In this embodiment, the weighted blending unit 216 operates to weighted blend (or interpolate) pixel information along the selected optimal angle Opt_Angle and pixel information along the normal axis, so as to render the pixel to be interpolated. The pixel information along the optimal angle and the normal axis is weighted according to certain weighting algorithm, and in this embodiment, a two-phase weighting algorithem is used in the weighted blending unit 216. Please note that, in order to clearly de-

scribe the weighted blending operation, the optimal angle Opt_Angle is assumed to have a value of 45 degrees. That is, the optimal angle is selected corresponding to a virtual axis extending along a line connecting the pixels Up[1] and Down[-1].

[0097] Before detailing such two-phase weighting algorithem, the following parameters are first defined:

[0098] Normal_Average_Pixel = (Up[0] + Down[0])/2

[0099] Angle_Average_Pixel = (Up[1] + Down[-1])/2

[0100] Normal_Pixel_Diff = |Up[-1] - Down[-1]| + |Up[0] - Down[0]| + |Up[1] - Down[1]|

[0101] Diff = |Angle_Pixel_Diff - Normal_Pixel_Diff|

[0102] wherein the parameter Normal_Average_Pixel is an average result of a plurality of pixel values along the normal axis, the parameter Angle_Average_Pixel is an average result of a plurality of pixel values along the optimal angle, and the parameter Diff is the absolute difference of the parameters Angle_Pixel_Diff and Normal_Pixel_Diff. It should be appreciated by those of ordinary skill in the art, that the averaging algorithms shown above serve merely as an example, and other averaging schemes may be used when the designer finds suitable.

[0103] Please refer to Fig.5, which illustrates a flowchart of a weighted blending algorithm adopted by the weighted blending unit 216 according to an embodiment of the present invention. According to Fig.5, the following steps are performed:

[0104] Step 502: Start the weighted blending algorithm;

[0105] Step 504:Determine a first weighting factor Weight_1. In this embodiment, Weight_1 is determined by comparing the parameter Angle_Pixel_Diff with a lower threshold value Thl_Down_1, with a nominal value of 20, and an upper threshold value Thl_Up_1, with a nominal value of Normal_Pixel_Diff, as follows:

[0106] If (Thl_Down_1

$\geqq$

Angle_Pixel_Diff)

[0107] Weight_1 = 0

[0108] Else if (Angle_Pixel_Diff

$\geqq$

Thl_Up_1)

[0109] Weight_1 = 1

[0110] Else

[0111] Weight_1 =

$$\frac{Angle\_Pixel\_Diff - Thl\_Down\_1}{Thl\_Up\_1 - Thl\_Down\_1}$$

[0112] Step 506: Calculate an intermediate interpolated pixel value Interpolation_Pixel_1 using Norma_Average_Pixel and Angle_Average_Pixel according to the first weighting factor Weight_1, as follows:

[0113] Interpolation_Pixel_1 = Normal_Average_Pixel × Weight_1 + Angle_Average_Pixel × (1 – Weight_1)

[0114] Step 508: Determine a second weighting factor Weight_2. In this embodiment, Weight_2 is determined by comparing the parameter Diff with a lower threshold value Thl_Down_2, with a nominal value of 0, and an upper threshold value Thl_Up_2, with a nominal value of 100, as follows:

[0115] If (Thl_Down_2

$\geqq$

Diff)

[0116] Weight_2 = 0

[0117] Else if (Diff $\geq$ Thl_Up_2)

[0118] Weight_2 = 1

[0119] Else

[0120] Weight_2 =

$$\frac{Diff}{Thl\_Up\_2 - Thl\_Down\_2}$$

[0121] Step 510: Calculate an output interpolated pixel value Interpolation_Pixel_2 using the intermediate interpolated pixel value Interpolation_Pixel_1 and the parameter Angle_Average_Pixel according to the second weighting factor Weight_2, as follows:

[0122] Interpolation_Pixel_2 = Interpolation_Pixel_1 × (1 − Weight_2) + Angle_Average_Pixel × Weight_2

[0123] Step 512: Finish the weighted blending algorithm.

[0124] The output interpolated pixel value Interpolation_Pixel_2 is then transmitted as output to the succeeding circuitry. As a result of the aforementioned embodiment of the in-

vention, an intra-field interpolation method and apparatus have been provided to more adequately account for the intra-field interpolation mechanism in the de-interlacing application, as well as in other image processing applications.

[0125] Those skilled in the art will readily observe that numerous modifications and alterations of the device may be made while retaining the teachings of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.